

### Day 85

Q. You are given an undirected graph with  $N$  nodes (numbered 1 through  $N$ ) and  $M$  edges. Each edge connects two distinct nodes. However, there may be multiple edges connecting the same pairs of nodes, and they are considered to be distinct edges. A lowercase English letter is written in each node. You are also given a string  $S$  with length  $L$ . A beautiful path is a sequence of  $L-1$  edges such that there is a sequence of  $L$  nodes with the following properties:

for each valid  $i$ , the  $i$ -th edge connects the  $i$ -th and  $(i+1)$ -th of these nodes

for each valid  $i$ , the  $i$ -th character of  $S$  is written in the  $i$ -th of these nodes

There are no other restrictions — a path may visit nodes or edges any number of times in any order.

Determine the number of beautiful paths in the graph. Since the answer can be very large, compute it modulo  $(10^9)+7$ .

#### Input

The first line of the input contains a single integer  $T$  denoting the number of test cases. The description of  $T$  test cases follows.

The first line of each test case contains three space-separated integers  $N$ ,  $M$  and  $L$ .

The second line contains a single string  $S$  with length  $L$ .

The third line contains a single string with length  $N$ . For each valid  $i$ , the  $i$ -th character of the string is the letter written in the  $i$ -th node.

Two lines follow. The first line contains  $M$  integers  $u_1?, \dots, u_m?$ . The second line contains  $M$  integers,  $v_1?, \dots, v_m?$ . This denotes that there is an edge connecting nodes  $u_i?$  and  $v_i?$ . These edges are distinct, even though they may connect the same pair of nodes.

#### Output

For each test case, print a single line containing one integer — the number of beautiful paths modulo  $(10^9)+7$ .

#### Sample Input

```
2
4 4 3
aac
aaca
1 2 2 1
2 3 4 2
2 1 2
aa
aa
1
2
```

#### Sample Output

```
3
1
```

main.py

```
t = int(input())
mod = 1000000007

for _ in range(t):
    n, m, l = list(map(int, input().split()))
    s = input()
    v = input()
    e1 = list(map(int, input().split()))
    e2 = list(map(int, input().split()))
    adj = [[] for _ in range(n + 1)]
    count = {}

    for i in range(m):
        adj[e1[i] - 1].append(e2[i] - 1)
        adj[e2[i] - 1].append(e1[i] - 1)
        temp = [e1[i]-1, e2[i]-1]
        temp.sort()

        if tuple(temp) not in count:
            count[tuple(temp)] = 0
        count[tuple(temp)] += 1

    dp = [[0 for _ in range(l)] for _ in range(n)]
    for j in range(0, n):
        if s[0] == v[j]:
            dp[j][0] = 1

    for j in range(1, l):
        for i in range(0, n):
            if s[j] != v[i]:
                continue
            for k in adj[i]:
                dp[i][j] = (dp[i][j] + dp[k][j - 1]) % mod

    ans = 0
    for i in range(0, n):
        ans = (ans + dp[i][l - 1]) % mod

    if min(s) == max(s):
        for i in range(0, n):
            for j in range(i + 1, n):
                if v[i] == v[j] and (i, j) in count:
                    ans = (ans - pow(count[(i, j)], l - 1, mod) + mod) % mod

    print(ans)
```

output

```
2
4 4 3
aac
aaca
1 2 2 1
2 3 4 2
2 1 2
aa
aa
1
2
1
PS E:\Panku\Python> █
```